Attack on Blob: Mega Multiply

Renegadeware

V 1.0.0

# Table of Contents

(fill after)

# Game Overview

## Target Learning Objective

## [4.NBT.B.5 – Multiply Four by One and Two by Two Digit Numbers](https://drive.google.com/file/d/1f6soq0DJodacNKyBln6oD8dmowyJS9B8/view)

## Demographics

* Ages 8-11

## Genre, Theme, and Setting

* Arcade-Puzzle game with a scoring/grading system on how efficient and well (based on no. of mistakes) the player performs.
* Earth is in grave danger as space blobs have appeared (once more) in the night sky. Bigger and fatter than ever, these interstellar menaces have blotted the entire sky. Fast approaching, the player must banish them with the many secret techniques of multiplying large numbers.
* The game takes place in view of the starry night, with out of this world (literally) visuals to indicate the alien nature of the blobs (think the 70’s colorful-scheme and wavey-flow patterns).
* These blobs have a nonchalant look about them, sometimes with mirth, regardless of their own predicament.

## Core Gameplay

The game is broken up into two parts: lesson and blob banishment. Each lesson will introduce a concept on how to handle multiplying certain number of digits in the multiplication, and then followed by the blob banishment that puts those lessons into practice.

### Lesson

This is mostly non-interactive with explanation on how to deal with multiplying a certain number of digits on each factor. There will be a basic interactive tutorial on how to banish blobs, and using distributive principle (via area visual) to input the correct answer. Another basic interactive tutorial will be introduced later to deal with multiplying two two-digit numbers.

Once the lesson has been completed, the player can click on the “next” button to proceed to blob banishment. (Lesson archives could also be put here for player to review, tentative).

### Blob Banishment

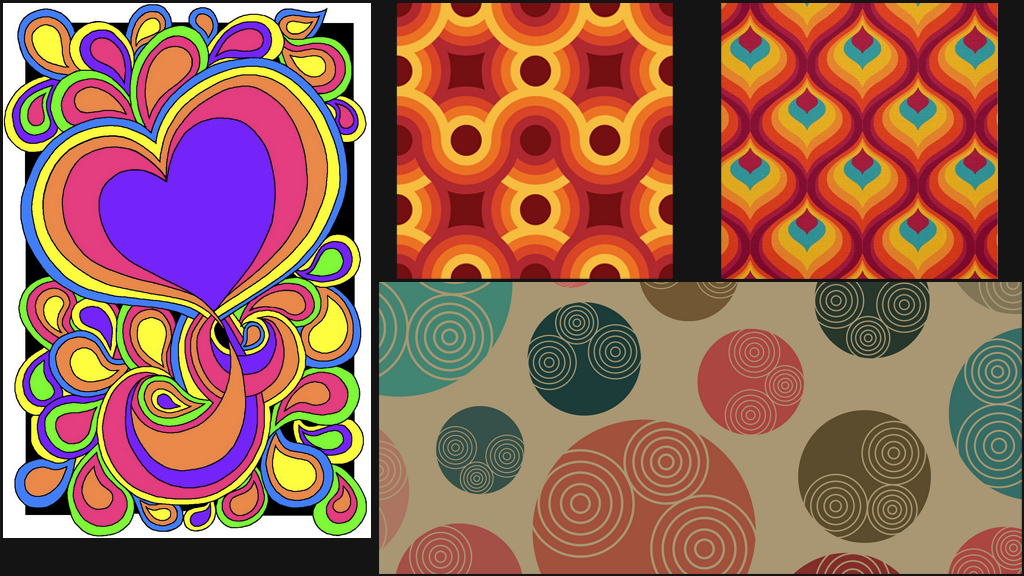
During blob banishment, the player is tasked with creating an attack blob that will eradicate a pair of blobs. This is where the player makes use of distributive principle, to be able to compute the correct product of the connected blob factors. Once all blobs are banished, the player can move on to the next lesson.

## Look and Feel

* 70’s color-scheme and wavey-flow patterns as backdrops, along with a starry night sky during blob banishment.
* Soothing animated background during lessons, with a reassuring robot that explains the lesson.
* Non-menacing, happy-go-lucky blobs with a simple beady-eye, along with a line-mouth showing expressions depending on the situation: happy, nonchalant, worried, joy. Distinct look for large and small blob to distinguish who can pair with who.

Examples of these can be seen from a previous game:

Attack on Blob: Multiply and Divide.



Various 1970’s patterns.

## Target Platform

* **WebGL with iPad support (iPad 6)** – The game will be completely mouse/touch driven, with the option to use the numpad/numbers from the keyboard when filling in the numbers.

# Game Flow

## Blob Banishment

### Summary

* 4 (or 5) blobs are spawned in an enclosed circular board. Two large distinct blobs, and two smaller distinct blobs. Each blob displays a number inside them.
* The player can connect a large blob to a small blob by dragging them from one to the other. Restrictions are made on which blobs can be connected (e.g. three digit numbers can only be connected to a one digit number, and vice-versa). There will be a highlight on which blobs can be connected when a blob is dragged.
* Once two blobs are connected, the attack commences with the following phases through a UI:

1. *Distributive* – visualizes the multiplication as the width and length of a rectangle, the player can click each digit in the 1’s, 10’s, etc. Once the player is satisfied with the distribution, they can press “evaluate” to proceed.
2. *Evaluate* – The player will need to solve each area created during the distributive phase (via a numpad UI). Once all the values are put in correctly, the player will be able to press “sums” to proceed.
3. *Sums* – All values are aligned vertically, with slots at the bottom representing each digit. The player will need to compute each digit from right to left (via a numpad UI). Any carry-over numbers will visually be added at the top of the next column to the left. Once all digits are put in correctly, the player will be able to press “attack” to proceed.

* Once the player successfully completes the attack, an attack blob will appear on the board with the correct product value for the paired blobs.
* The attack blob will automatically connect with the paired blobs, and all three of them will vanish.
* A combo counter will appear on the screen (or incremented if it appeared previously), the whole operation will be displayed at the top as a toast-notification, and the score is updated.
* Two more will appear, and the round counter is updated. Repeat the entire process until there’s no blob left.
* On some levels, after the first, anomaly blobs can appear on the board.
  + These have a more “tricky” values in them, and must be done with no mistake.
  + They vanish if you pair other blobs, or if you make a mistake while pair a blob to it.
  + Successfully banishing them will also eliminate two other blobs on the board.
  + To break the possible monotony of gameplay, different problem solving may occur during the attack phase of an anomaly blob: no distribute phase, but directly need to fill in values in sums phase; find the erroneous products instead of filling the values; etc. Will need to iterate on this to see what is possible.
* Victory fanfare, and then a summary UI is shown with scores on each category of the player’s performance, along with a grade. (Note: could also put a retry here if they got a low grade).
* Player clicks on the “next” button to proceed to the next lesson.

### Mechanics

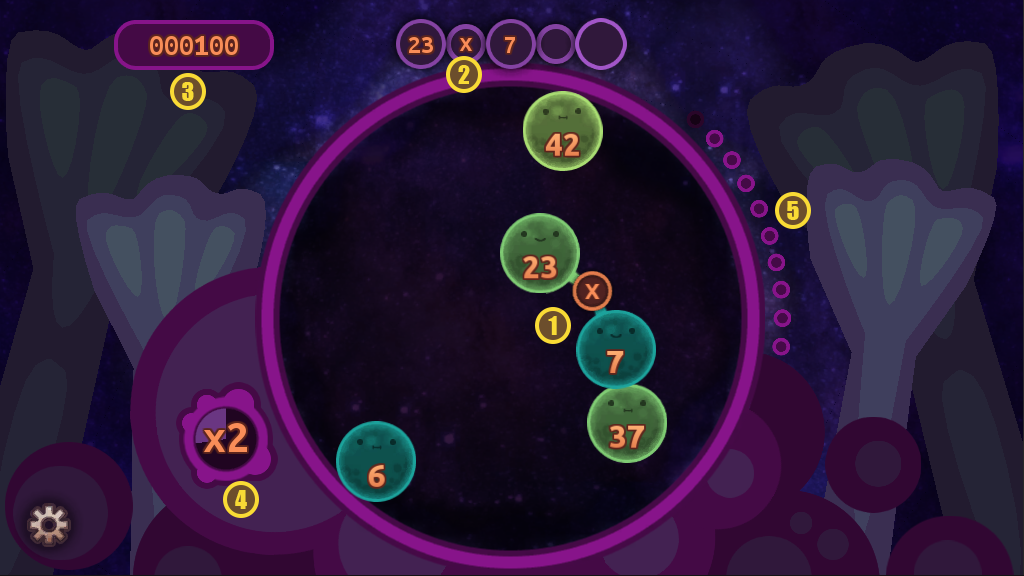
* During the blob connecting phase, the player will be able to choose which values to multiply. The player can take advantage of this by connecting values they are most comfortable, or save the easier values for later.
* In the Distributive phase of the attack, the player simply clicks on the appropriate digits to split the values up. Once the player is comfortable with the way they’ve distributed the factors, they can proceed to evaluate each area.
* The player will be using a numpad to input the values for the products during Evaluation, as well as each digit in Sums.

### Losing Gameplay / Incorrect Concept Understanding

* During the attack, there will be a visual counter indicating how many tries the player have left before they are booted back to connecting blobs.
* A visual/sound will occur if the player incorrectly inputs a value during “Evaluate” and “Summation”. This will decrement the visual counter.
* Once the visual counter reaches zero, the player is booted out of the attack, and will need to reconnect a new pair of blobs.
* The summary UI in the end will show how many errors the player made during the attack phase: Evaluate and Summation. A grade will be displayed to summarize the player’s overall performance.
* If the player gets a grade: ‘C’ or below, they can opt to retry the level (maximum: 2).

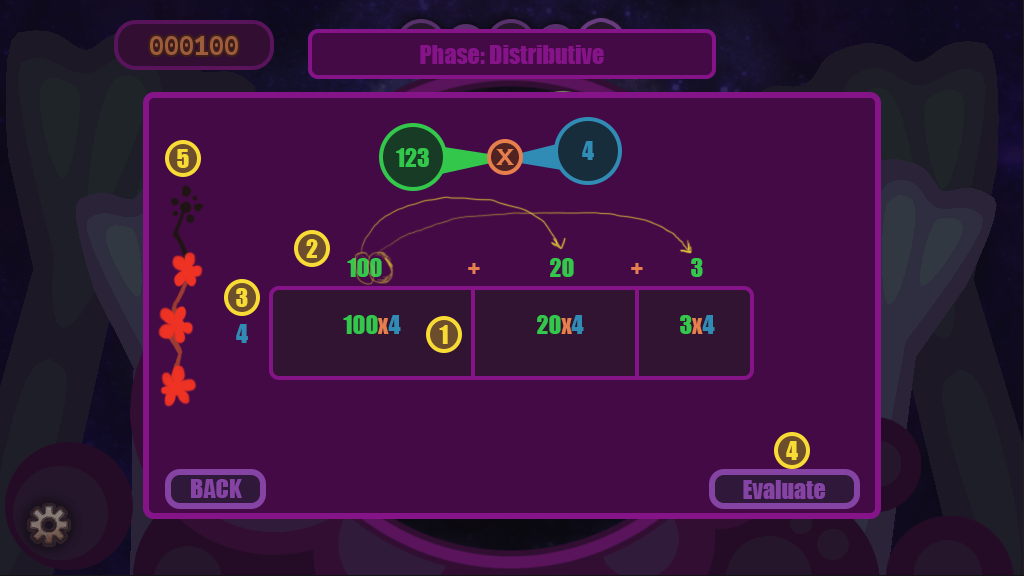
### Mockups

Blob Connecting



|  |  |  |
| --- | --- | --- |
| 1 – Players drag one blob to another to connect. | 2 – Displays the equation based on the connected blobs. Eventually shows the product after the entire attack phase. | 3 – Current score accumulated. |
| 4 – Score multiplier. Increments if the player does an attack correctly. Decrements each time an attack fails (loses all tries within attack phase). | 5 – Rounds counter. This tells the player how many more blobs to banish. |  |

Attack Phase: Distributive



|  |  |  |
| --- | --- | --- |
| 1 – Area multiplications are shown here as the player splits the digits of either factors of the equation. | 2 – First factor that is represented as the width of the area. The player can click on valid digits to split them up. An animation will display this process, and the area will split up with a new column. | 3 – Second factor that is represented as the height of the area. Just like the first factor, the player can split the digits up to create new rows. This is only interactable when the player is multiplying two two-digits numbers. |
| 4 – Once the player is satisfied with splitting up the factors, they can click this to evaluate each area. | 5 – Attack Hit points. This shows how many more mistakes the player can make. If it reaches zero, they are booted out of the entire attack phase. |  |

Attack Phase: Evaluate Areas



|  |  |
| --- | --- |
| 1 – Completed product. Once the player inputs the correct product, the cell will be locked with a ‘completed’ visual. | 2 – Cells that still need the correct value can be clicked on to bring up the numpad for the player to input the product. |
| 3 – Numpad. Displays the current equation the player must solve. Entering the incorrect value will cause a ‘wrong’ effect and decrement the player’s attack hit points. (There should be a ‘close’ button here to back out) | 4 – Once all cells are evaluated correctly, this button will be enabled, and the player can proceed to the final attack phase. |

Attack Phase: Sums



|  |  |
| --- | --- |
| 1 – List of all products to be added. The highlighted column is the current digit the player must compute. The order goes from right to left. | 2 – These are the digits of the final product. Clicking on it will show the numpad where the player must enter the correct value. The values shown at the top of the numpad indicates the sums of digits the player is computing. |
| 3 – Carryovers are displayed here if the value of the previous digit is > 9. There will be a visual animation showing this process, to let the player know there’s one additional value to consider. | 4 – Once all digits are correctly filled, the player can press this to finally form the banish blob with the final product value. |

Blob Banishment



|  |  |
| --- | --- |
| 1 – Once the final product is completed, the banish blob will appear and connect with the paired factors. All three will vanish with some fanfare. New blobs will spawn if there are still some available in reserve. | 2 – The solved equation pop-up. |
| 3 – Combo is incremented upon a successful banishment. | 4 – Score is updated based on the player’s performance, factoring in the combo multiplier. |

Victory UI



## Part 1 – Multiplying 2 digits with 1 digit.

### L.O. Concepts Covered

* Multiply a whole number of two digits by a one-digit whole number using strategies based on place value.
* Multiply a whole number of two digits by a one-digit whole number using strategies based on the properties of operations.
* Multiply a whole number of two digits by a one-digit whole number using strategies based on place value and the properties of operations.
* Represent multiplication using arrays.

### Summary

* The lesson part of the game will start with a quick explanation of distributive property and how to visualize multiplication using the area of a rectangle.
* Tutorial on the ‘blob banishment’ gameplay. The player is guided throughout the process.
* The blob banishment will comprise of 2-digits and 1-digit factors.

## Part 2 – Multiplying 3 digits with 1 digit.

### L.O. Concepts Covered

* Multiply a whole number of three digits by a one-digit whole number using strategies based on place value.
* Multiply a whole number of three digits by a one-digit whole number using strategies based on the properties of operations.
* Multiply a whole number of three digits by a one-digit whole number using strategies based on place value and the properties of operations.

### Summary

* The lesson part will further explain the distributive property, but now with a 3-digits factor.
* The blob banishment will comprise of 3-digits and 1-digit factors.

## Part 3 – Multiplying 4 digits with 1 digit.

### L.O. Concepts Covered

* Multiply a whole number of four digits by a one-digit whole number using strategies based on place value.
* Multiply a whole number of four digits by a one-digit whole number using strategies based on the properties of operations.
* Multiply a whole number of four digits by a one-digit whole number using strategies based on place value and the properties of operations.

### Summary

* The lesson part will further explain the distributive property, but now with a 4-digits factor.
* There will also be a lesson about 0’s within the digits that the player must be careful with when doing partial products.
* The blob banishment will comprise of 4-digits and 1-digit factors.

## Part 4 – Multiplying 2 digits with 2 digits.

### L.O. Concepts Covered

* Multiply two two-digit numbers using strategies based on place value and the properties of operations.
* Explain multiplication of a whole number up to four digits by a one-digit whole number and of two two-digit numbers by using equations, rectangular arrays, and/or area models.

### Summary

* The lesson part will explain how to handle multiplying two 2-digits factors by expanding the area model with an extra row.
* The player can now split the digits of both factors.
* The blob banishment will comprise of 2-digits factors. All blobs can be matched with each other since they all have the same number of digits.

## Part 5 – Score Reviews before the end.

### Summary

* Allow players to review the scores they made across all the levels in the game, along with the total score and rank.
* This is here for people to see how well they did, and see where they may need to improve their understanding of the concepts covered by the game.
* They can press the “next” button to proceed to the end, where an end scene is played, and the game concludes.

# LO Concept Coverage

## Academic Concepts

* Multiply a whole number of up to four digits by a one-digit whole number using strategies based on place value.
* Multiply a whole number of up to four digits by a one-digit whole number using strategies based on the properties of operations.
* Multiply two two-digit numbers using strategies based on place value and the properties of operations.
* Multiply a whole number of up to four digits by a one-digit whole number using strategies based on place value and the properties of operations.
* Explain multiplication of a whole number up to four digits by a one-digit whole number and of two two-digit numbers by using equations, rectangular arrays, and/or area models.
* Represent multiplication using arrays.

# Legends of Learning Required Content Practices

## Checklist Overview

|  |  |
| --- | --- |
| **ITEM** | **COVERED** |
| Players should learn and be held accountable through gameplay-based problem solving and experience. Players should not be learning primarily through text-based instruction or assessment items. |  |
| Game does not include multiple choice assessment items. |  |
| All instruction is scientifically and mathematically correct. |  |
| Confirm that the game is linked to 2/3 or 5 main concepts of the total, whichever is greater. Confirm that the linked main concepts are correctly covered in the game. |  |
| All on-screen words spelled correctly and grammatically correct. |  |
| Vocabulary and reading level appropriate for the lowest grade level within the target audience and grade band. |  |
| Game does not include material that is inappropriate for school. This includes, but is not limited to: violence, firearms, bombs, knives, daggers, blood, gore, smoking, vaping, drug use, any mind-altering substances, alcohol, harm to human-looking characters, harm to animals, insinuating killing or death, ideally they’re always chased away rather than eliminated. If there is conflict with an enemy in game, they are chased away rather than eliminated or killed. (There can be death if it is in the context of the learning objective – ex. The food chain) If you have any questions about this policy and your game, please ask us. |  |
| Game avoids any stereotypic presentation of gender, race, region, or culture. |  |
| Characters are diverse in gender, race, culture, and ability. |  |
| Players cannot simply click through and complete the game without learning. Players should be prompted to re-learn and re-do portions of the game where they had poor results due to less understanding of the academic material. Avoid the word “FAIL” if the student incorrectly understands academic material. |  |
| Academic problems are not consistently repeated. Players are presented with different problems to solve. |  |
| Gameplay mechanic reinforces the academic material, rather than being completely separate from instruction. I.e, there is a focus on academic reasoning rather than concept / question repetition. |  |
| Gameplay is intuitive and a player in the target age range can navigate the game and beat it with enough effort. |  |
| Games should be fun and interesting, designed as non-educational games are designed, with design to encourage players to keep playing. |  |
| Game is between 5 and 25 minutes in duration. |  |
| All text must be large, clear and concise with font sizes that can be read on a small Chromebook screen. |  |

## Connection Between Gameplay and Learning

* During the attack phase, the player is directly involved with distributing the factors. The visualization of the process will help the student understand the relationship of each digit of a number, as well as how they are represented as products of the total area.
* Players are tasked to evaluate each area of the distributed factors, thus assessing their aptitude in multiplication.
* Players are also tasked to add all evaluated area to form the final product of the equation. They are tasked to evaluate each digit to assess how they understand carryovers when a summation of a digit is > 9.

## Role of Text in Learning

* A reassuring robot will be talking to the player to explain the various tricks of multiplying large numbers.
* Many of the explanation will be displayed in dynamic illustrations. Animations will help the player understand the properties of distribution, digits, and how they can be represented as an area.
* A step-by-step process will also help the player understand the game mechanics, as well as reinforce the lessons they just learned (guided by the robot).

## Characters – Diversity

* A reassuring robot talks throughout the game, it is gender neutral.
* All blobs are presented in a variety of sizes and colors, with a cheerful demeanor.

# Technical

## Development Hardware/Software

* All developers are using a PC with a Windows 10 OS.
* iPad 6 for tablet testing.
* Browsers: Firefox, Edge, Chrome, and Safari (via iPad 6)
* Game engine: Unity 2020.3.44+

## Asset Summary

* There will be some borrowed assets from the previous project: Attack on Blob: Multiply and Divide.
* Some logic from Attack on Blob: Multiply and Divide will be reused as well, but with the big addition of a new attack phase for the purpose of this subject’s LOs.
* Blobs will have the benefit of upgraded art to accommodate for various sizes.
* Game will be delivered via WebGL (targeting desktops and touchpads).

## Music and Sounds

* Will most likely use some music from Kevin Macleod, all under the [Creative Commons Attribution license](https://creativecommons.org/).
* We have a growing library of sound effects that are comprised of public licenses, as well as purchased licenses.

# Art Style

## Mockups

(show some made up stuff)

# Schedule for Development + Delivery

* Projected coding time: 2 – 3 weeks.
* Projected art asset time: 2 – 3 weeks.
* Projected lesson implementation time: 1 – 2 weeks.
* Projected polish time (intro, ending, etc.): 2 – 3 weeks.

# Story/Narrative

## Back Story

* Sequel to Attack on Blob: Multiply and Divide
* After some time, the collective blobs that are trapped in another dimension have started fusing together to form larger number values. This has caused an instability to the seal, causing it to break.
* This rupture of the seal, and spilling of bigger blobs will be shown in the intro of the game.

## Plot Elements

* The use of the attack blob is a new solution to the problem presented. Whereas in the previous game, they were all banished together into another dimension. This time around, each blob is banished by these mysterious blob beings that appeared.